**US Army Corps
of Engineers**
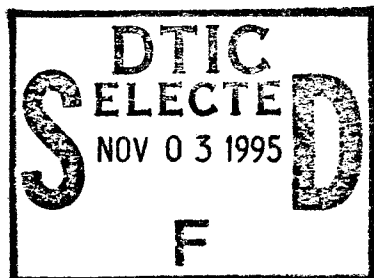Waterways Experiment
Station

Computer Applications in Geotechnical Engineering Program

Basic Structured Documentation to the Corps of Engineers National Dam Inventory Data Update Program Based on the E-R Diagram and Structure Chart

by   Alberto Tavares da Silva

DTIC
S ELECTE D
NOV 0 3 1995
F

19951102 030

DTIC QUALITY INSPECTED 5

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

Basic Structured Documentation to the Corps of Engineers National Dam Inventory Data Update Program Based on the E-R Diagram and Structure Chart

by   Alberto Tavares da Silva

U.S. Army Corps of Engineers
Waterways Experiment Station
3909 Halls Ferry Road
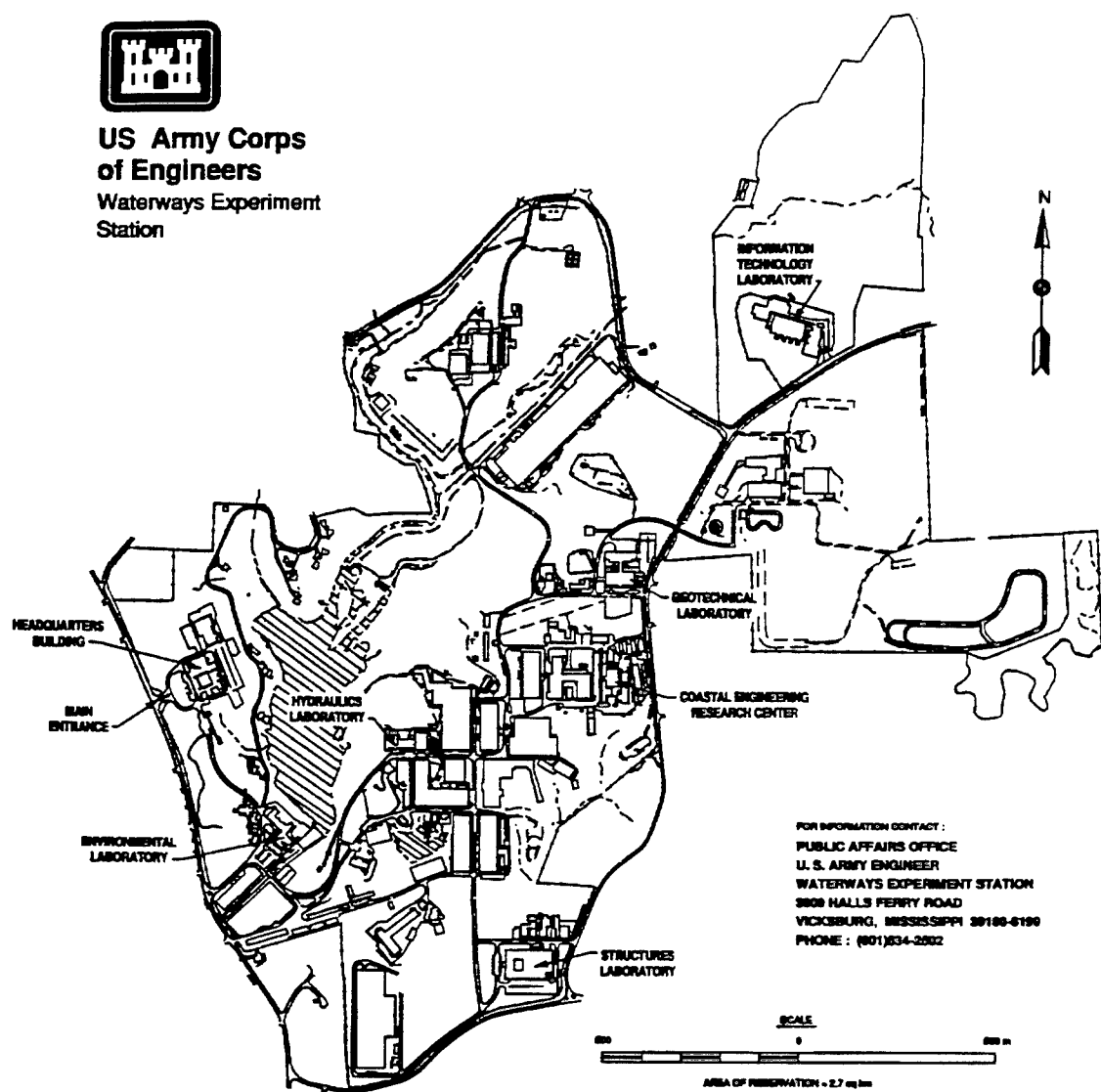Vicksburg, MS   39180-6199

Final report

Approved for public release; distribution is unlimited

Prepared for   U.S. Army Corps of Engineers
              Washington, DC   20314-1000

**US Army Corps
of Engineers**
Waterways Experiment
Station

INFORMATION
TECHNOLOGY
LABORATORY

N

GEOTECHNICAL
LABORATORY

HEADQUARTERS
BUILDING

HYDRAULICS
LABORATORY

MAIN
ENTRANCE

COASTAL ENGINEERING
RESEARCH CENTER

ENVIRONMENTAL
LABORATORY

FOR INFORMATION CONTACT :
PUBLIC AFFAIRS OFFICE
U. S. ARMY ENGINEER
WATERWAYS EXPERIMENT STATION
3909 HALLS FERRY ROAD
VICKSBURG, MISSISSIPPI 39180-6199
PHONE : (601)634-2502

STRUCTURES
LABORATORY

SCALE

AREA OF RESERVATION = 2.7 sq km

Contents

List of Figures

List of Tables

Preface

This report was prepared under the auspices of the Dam Safety Task Group, Computer Applications in Geotechnical Engineering (CAGE) Program, U.S. Army Corps of Engineers. The CAGE Program is managed by the U.S. Army Engineer Waterways Experiment Station (WES), Geotechnical Laboratory (GL), and is sponsored by Headquarters, U.S. Army Corps of Engineers (USACE). The USACE Technical Monitor is Mr. Art Walz, Directorate of Civil Works, Engineering Division, Geotechnical and Materials Branch. Chairman of the CAGE Policy Group is Mr. David P. Hammer, U.S. Army Engineer Division, Ohio River. Chairman of the Dam Safety Task Group is Mr. Jim Sekela, Geotechnical Branch, U.S. Army Engineer District, Pittsburgh.

Development of this report and the program modifications described herein were accomplished by Major Alberto Tavares da Silva, an Officer of the Brazilian Army Corps of Engineers, during an assignment to WES between August 1994 and August 1995 as part of a Brazilian and U.S. Army interchange. The work described in this report was completed within the Soil Mechanics Branch (SMB), Soil and Rock Mechanics Division (SRMD), GL. Chief of SMB at the time of the preparation of this report was Mr. Milton Myers. Dr. Don Banks was Chief, SRMD, and Dr. W. F. Marcuson was Director, GL. Ms. Wipawi Vanadit-Ellis of SMB was coordinator of the CAGE Program.

At the time of the publication of this report, Director of WES was Dr. Robert W. Whalin. Commander of WES was COL Bruce K. Howard, EN.

For further information concerning this report, contact Mr. Milton Myers, (601) 634-2640, or Ms. Wipawi Vanadit-Ellis, (601) 634-3183.

Summary

This report was developed by Major Alberto Tavares da Silva, a Brazilian Army Officer, working at Geotechnical Laboratory (GL), U.S. Army Engineer Waterways Experiment Station (WES), between August 1994 and August 1995 as part of a Brazilian and U.S. Army interchange.

Major Tavares has a civil engineer graduate degree and a computer science masters degree. His work at CEWES-GS-S includes instrumentation, monitoring, and safety of earth dams. This subject is a matter of Brazilian Army interest because of dam construction in the northeast of Brazil, a very dry interior area.

His computer science specialization area is that of software engineer. These two powerful software tools are applied to generate a basic structured documentation to the Corps of Engineers National Dam Inventory Data Update Program, developed at CEWES-GS-S by Mr. Earl V. Edris, Jr., Ms. Wipawi Vanadit-Ellis, and Ms. Kay Woo, GL.

This work was developed under the guidance of Mr. W. M. Myers, Chief, Soil Mechanics Branch, GL.

1 Introduction

This Structured Project is a universal methodology that enabled the development of a computer system within a life cycle as shown in Figure 1.
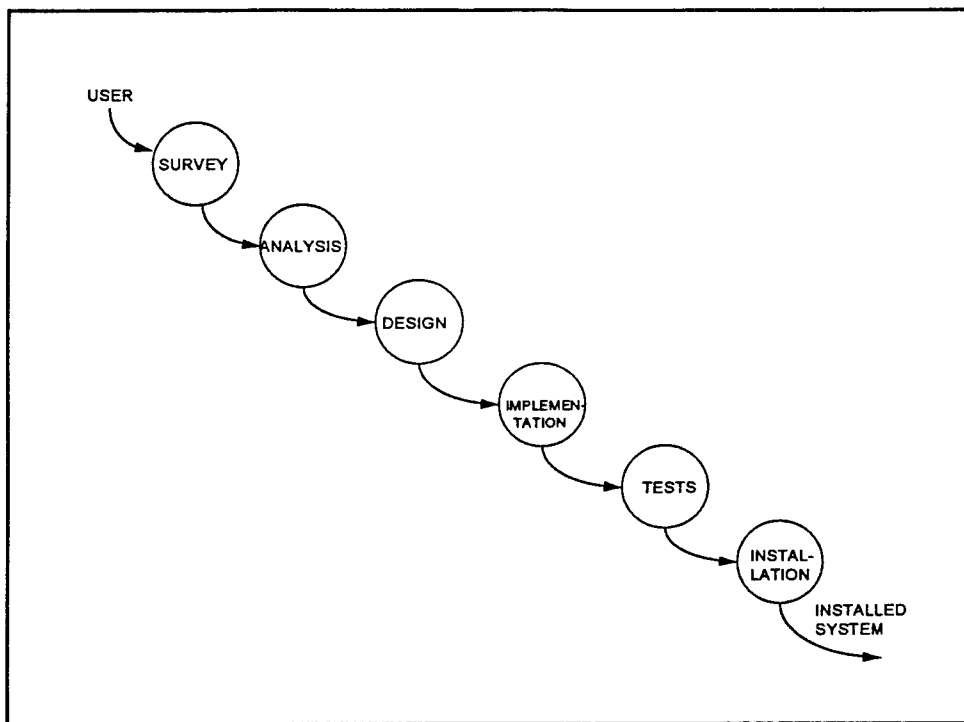
Figure 1.　System life cycle

A resume of each phase is:

- Survey.

At this phase, a preliminary study of the system is developed. The objective is to approve or disprove.

- Analysis.

Two methods can be applied at this phase: Essential Analysis and Structured Analysis. Both methods use the data flow diagram for function modeling and entity-relationship for data modeling. Each method has a logical abstraction level. The basic difference is in the analysis strategy, because the Essential Analysis uses the event concept.

- Design.

The tool used here is the Structure Chart. This chart has a physical abstraction level, since this phase proceeds the implementation of the system.

- Implementation.

The programs are codified.

- Tests.

The acceptance tests are realized.

- Installation.

The system is installed for operation and includes the all life system maintenance.

Following the life cycle is difficult for engineers other than computer engineers, because the documentation is very extensive and the maintenance time is very high. A minimal documentation is necessary to warrant a good understanding. Imagine one system with 3,000 code lines specified only through the source codes. The programmer will have no problem if he is familiar with the program. However, a programmer who is unfamiliar with the program can face difficulty with a change in some of the code lines. A minimal documentation is necessary for program maintenance.

2 Objectives

This report has the following objectives:

- It describes two basic tools for program development: Entity-relationship (E-R) Diagram and Structure Chart.

- It generates a basic structured documentation to the Corps of Engineers National Dam Inventory Data Update Program based on the E-R Diagram and Structure Chart.

- It presents some improvements to the Corps of Engineers National Dam Inventory Data Update Program based on the E-R Diagram and Structure Chart.

3 Entity-Relationship Diagram

The Entity-Relationship (E-R) diagram is a data modeling tool which generates the data stores or files that will be used by the programs and was developed by Peter Chen, author of *Data Base Management*. Its use is not mandatory as with some scientific programs that need no data stores.

The E-R Diagram is a powerful tool for database specification based on a relational model and is composed of entities, relationships, attributes, and keys.

Entity

This program has existing form sets that discern each other. Figure 2 shows the entity graphic representation.

    Ex: Real object---> place, instrument, dam, etc.
        Person   ---> engineer, student, etc.
        Abstract ---> organization, aptness, etc.

Attributes

Entity characteristics are attributes. Figure 3 shows the attribute graphic representation. It is unnecessary to place the attribute connections in the diagram, because an entity with many attributes can generate a legibility problem. It is only a didactic representation.

    Ex: Instrument (numberIns, name, height, weight, purpose, manufacturer)

Key

An entity is a set of similar things or people called elements, but the existence of two equal elements should not be justified in the same entity. A
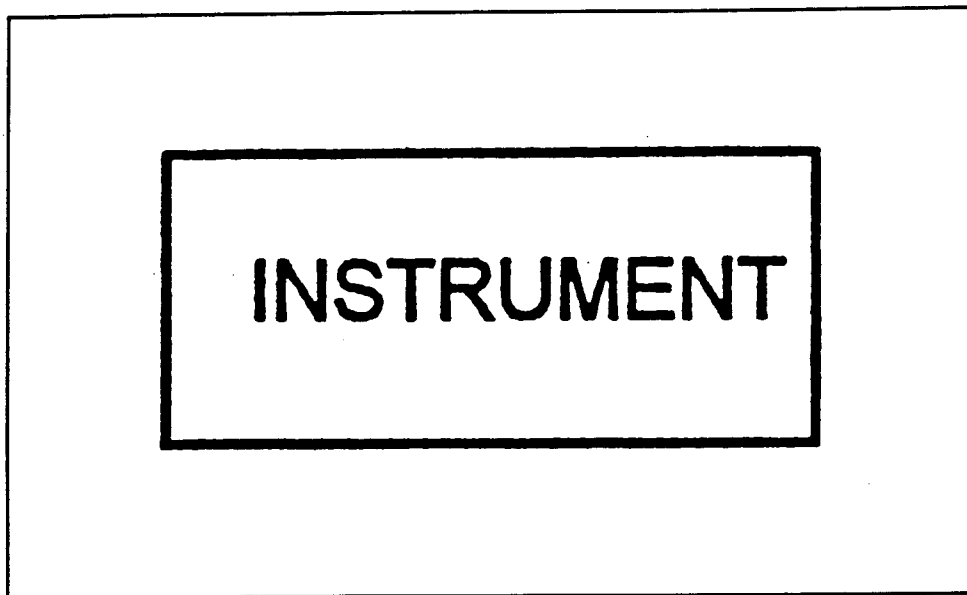
Figure 2.   Entity
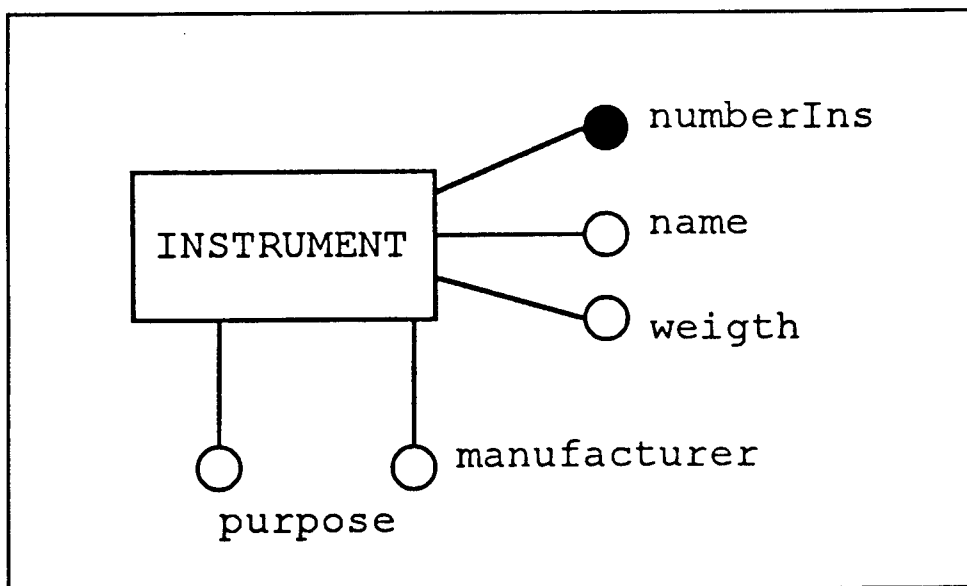
Figure 3.   Attributes

necessary mechanism called a key identifies one or more attributes that distinguish one element from all others. If the key is composed of one attribute, it is called a simple key; if composed of more than one attribute, it is called a compound key. In Figure 3, the attribute *numberIns* identify the elements of the Instrument entity.

The understanding of the key attribute explains one important concept called foreign key. A foreign key is a key attribute at one entity and an

attribute at another, as illustrated in the example below. It is a fundamental concept, because it generates a relational link between entities.

Foreign key example:

Dam(codeDam, name, city, high, length, capacity)

ey

foreign key

Instrument (numberIns, name, weight, manufacturer, purpos
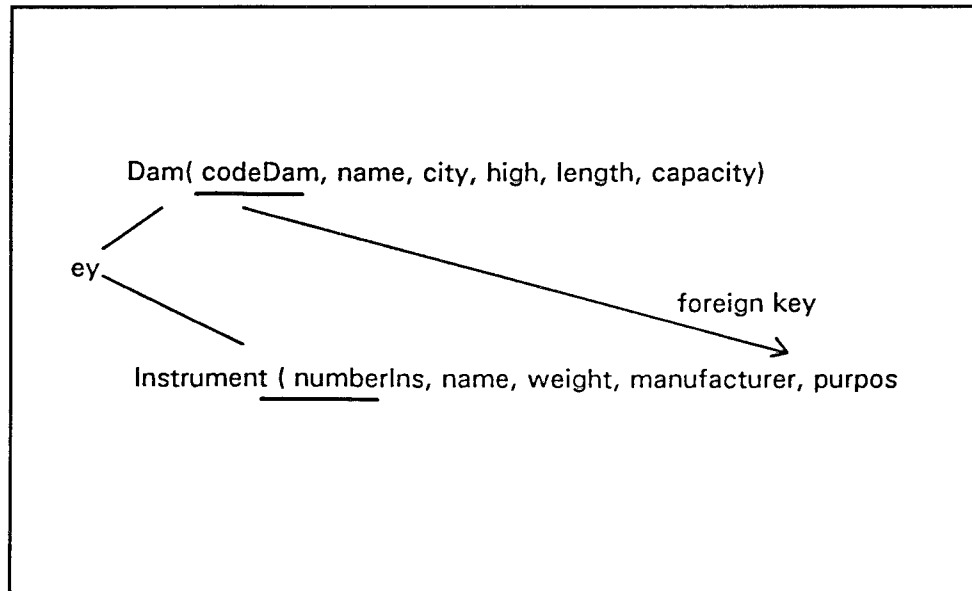
Relationship

A relationship represents one entity association. Figure 4 shows a relationship graphic representation where the entities DAM and INSTRUMENT are related by the relationship HAVE. The relationships can have their own attributes as illustrated in Figure 4.

Relationship types:

- 1:1

  Figure 5 shows that all occurrences in A correspond to 0 or 1 occurrence in B.

  Figure 6 shows a 1:1 relationship example between the entities ENGINEER and PROJECT, such as, one engineer works at only one project and the project has only one responsible engineer.

- 1:N

  Figure 7 shows that all occurrences in A correspond to 0 or N occurrences in B.

6
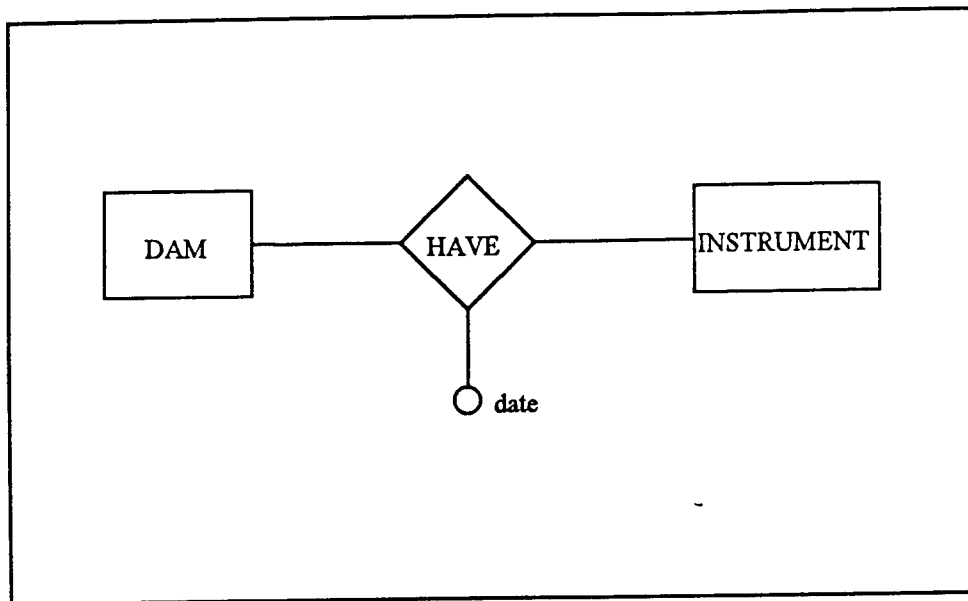
Figure 4.   Relationship
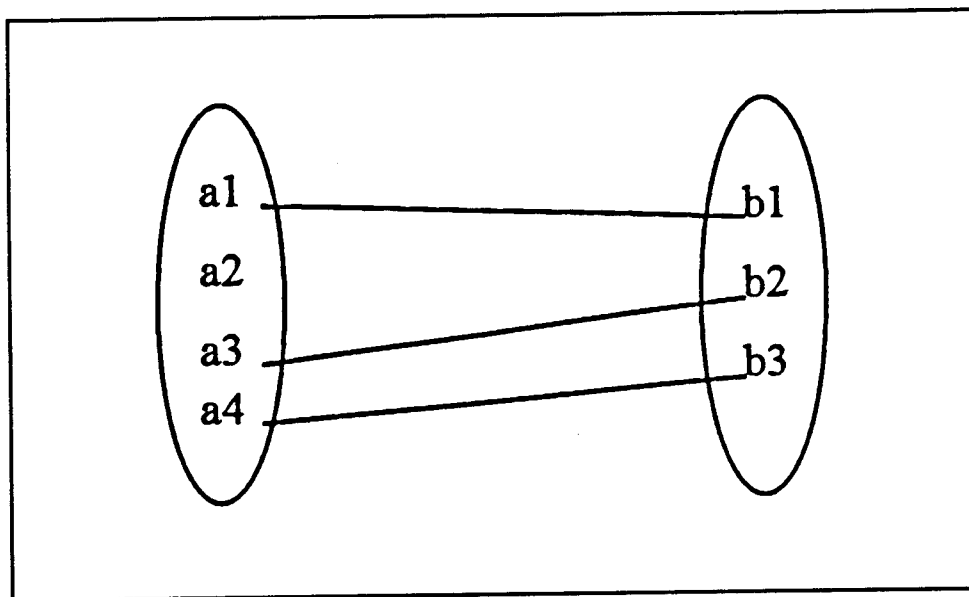
Figure 5.   Relationship 1:1

Figure 8 exemplifies a 1:N relationship. One dam has relations with
more than one element of INSTRUMENT, but each INSTRUMENT
element has relations with only one dam element. One dam can have
more than one instrument, but only one instrument can be installed at
a dam.
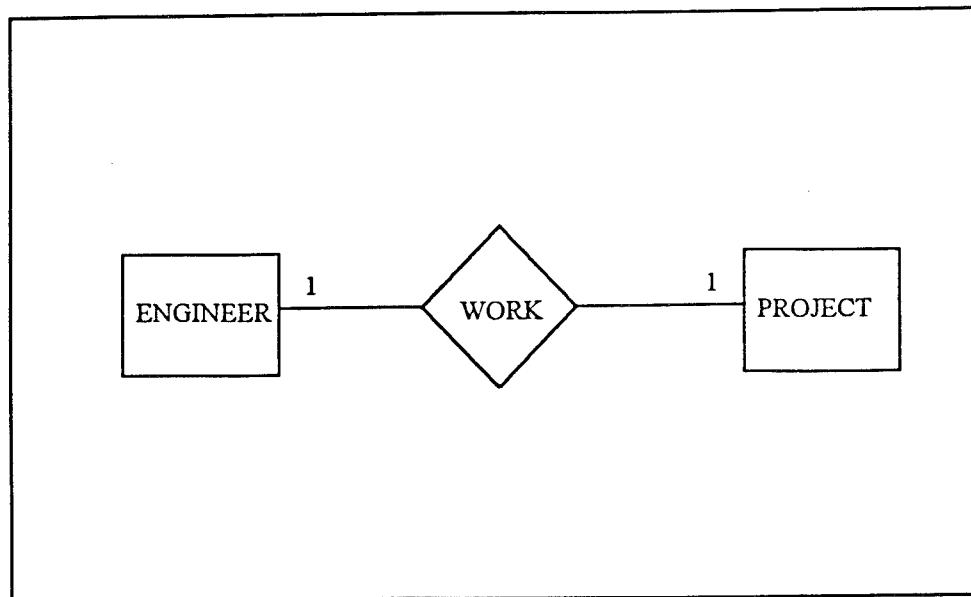
Figure 6.　1:1 relationship example
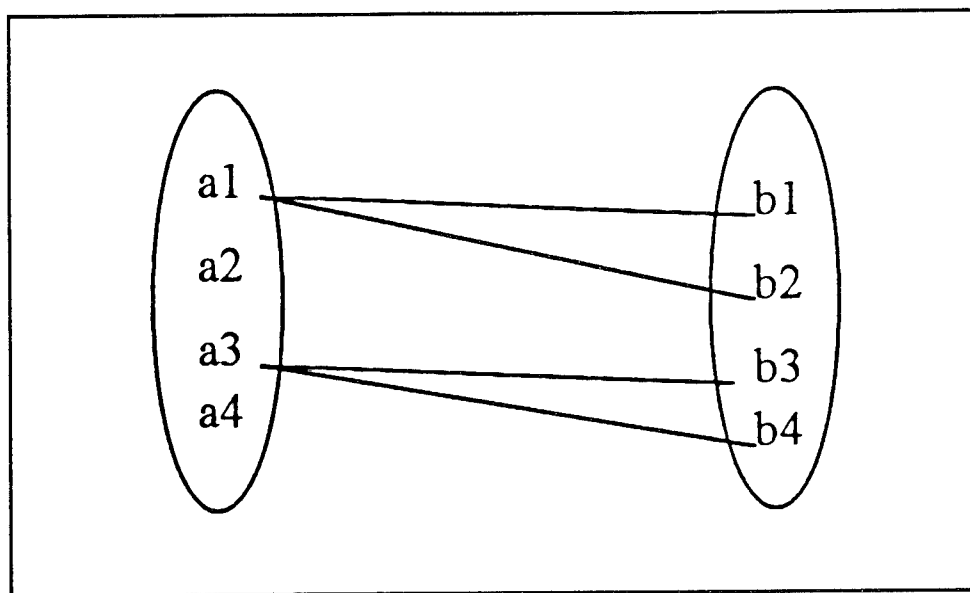
Figure 7.　1:N relationship

- N:M

  Figure 9 shows that all occurrences in A correspond to 0 or N occurrences in B and vice versa.
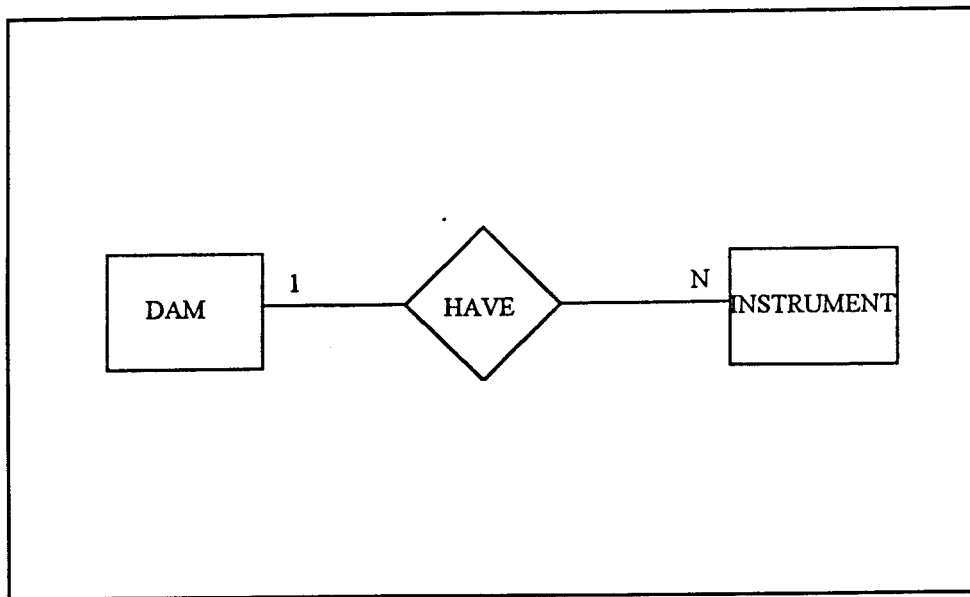
Figure 8.   1:N relationship example
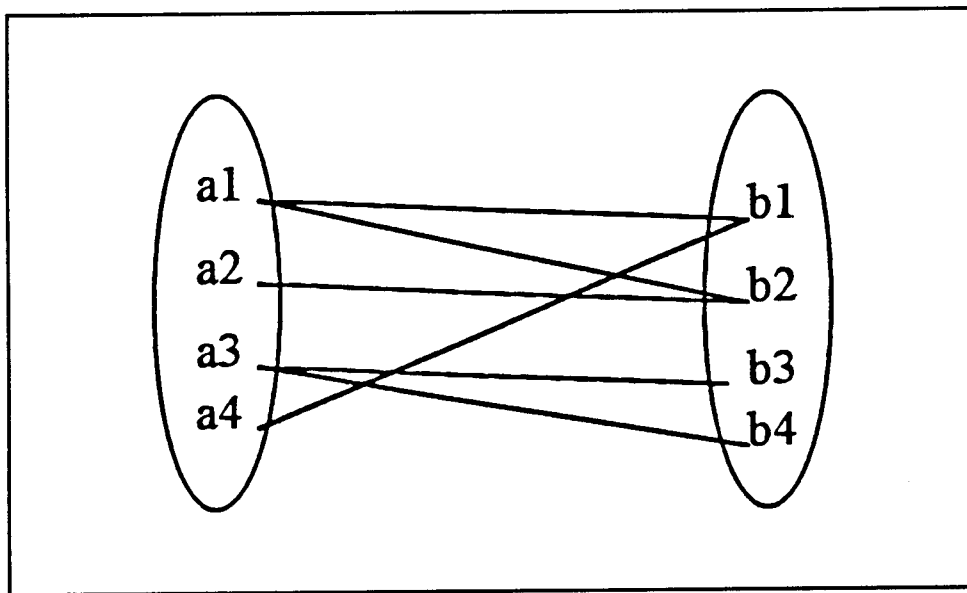
Figure 9.   N:M relationship

Figure 10 exemplifies a N:M relationship, i.e., one INSTRUMENT has a relationship with other elements of a MANUFACTURER and vice versa. Therefore, one instrument is produced by one manufacturer and other instruments are produced by other manufacturers.

● **N:M:P**

Figure 11 shows a ternary relationship between entities.
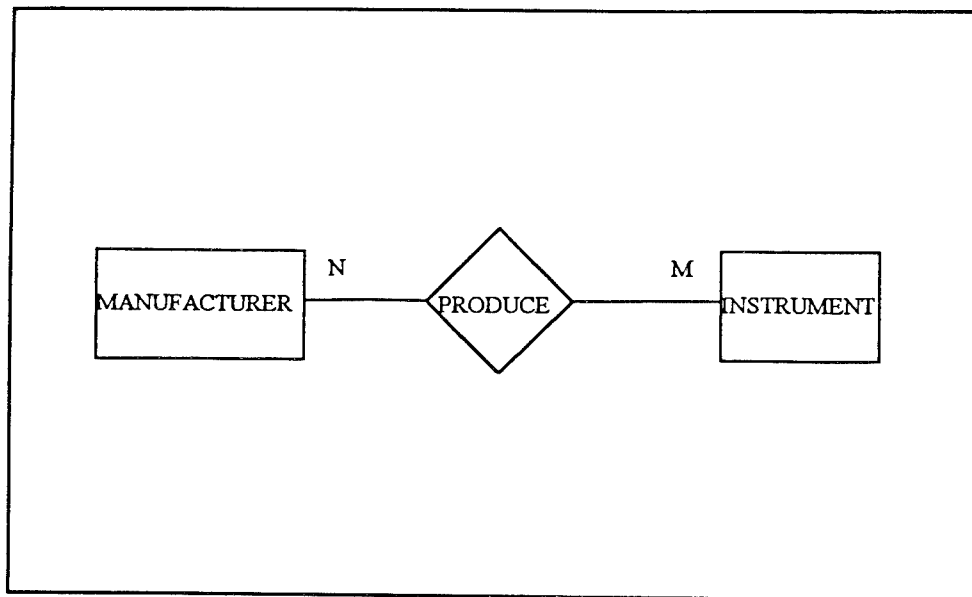
Figure 10. N:M relationship example
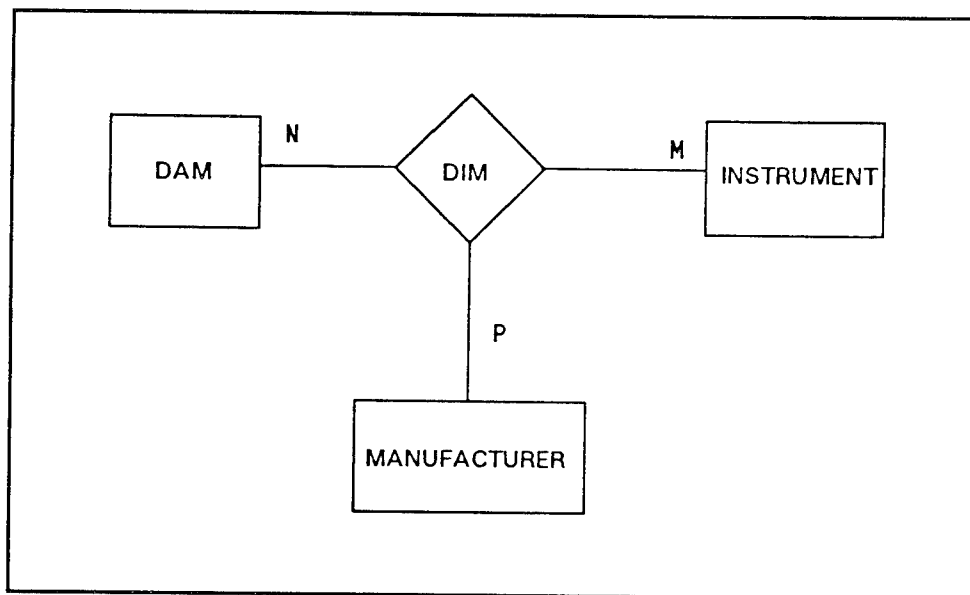
Figure 11. N:M:P relationship example

Files Generation

The E-R diagram generates a logical level specification. It is necessary to transform entities to files, in a physic level, that will be updated by the function of the program or by some interactive fourth-generation language. The file numbers generated will depend on the relationship between the entities.

- 1:1

  All relationships 1:1 generate two files, and the key attribute of either entity will be a foreign key in the other.

- 1:N

  Most of the relationship 1:N generates two files, and the entity with cardinality N receives the foreign key, according to the examples in this chapter. If the relationship has its own attributes, it is necessary to give special attention. The relation can generate each file with its own attributes with the keys of both entities related.

- N:M

  All relationships N:M generate three files: two files from the entities and one file from the relationship with its own attributes. The keys of both entities are related as shown in Figure 12.
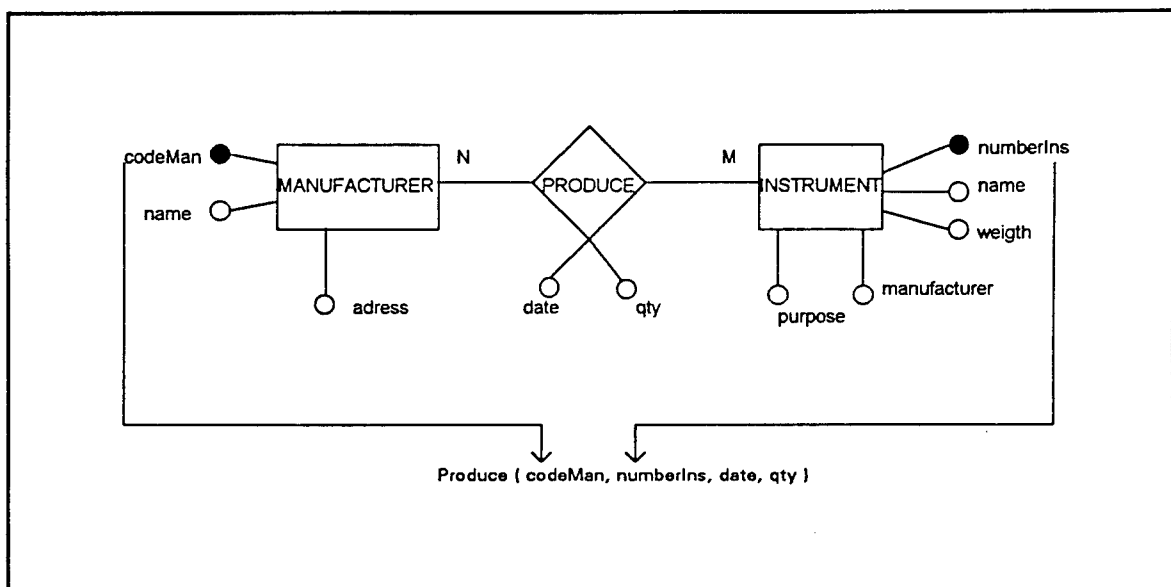
Figure 12. N:M relationship file generation

E-R Diagram Activities

It is important to list the activities that will support enable generation of the E-R Diagram:

- Entities definition.

- Relationship definition.

- Relationship types definition (1:1 / 1:N / N:M / N:M:P).

- Attributes and keys definition.

4   Structure Chart

A graphic tool permits decomposition of program modules, creates a hierarchy between them, and describes the program structure.

The components of a Structure Diagram are described as follows.

Modules

A module is the basic building block of the program. There are many kinds of modules. The most common are procedure and function. It can be defined as a program instructions set with five basic attributes:

- Name: identify the module.

- Input and output: input and output data.

- Function: what to do with the entrance data to produce the output data.

- Logic: logic which run the functions.

- Internal data: data referenced only by the own module.

It is very important that a previously written or a preexisting module can be stored in a library. Figure 13 shows the module graphic representations.

Connections

Connections represent the hierarchy between modules. Figure 14 shows module A calling module B.

13
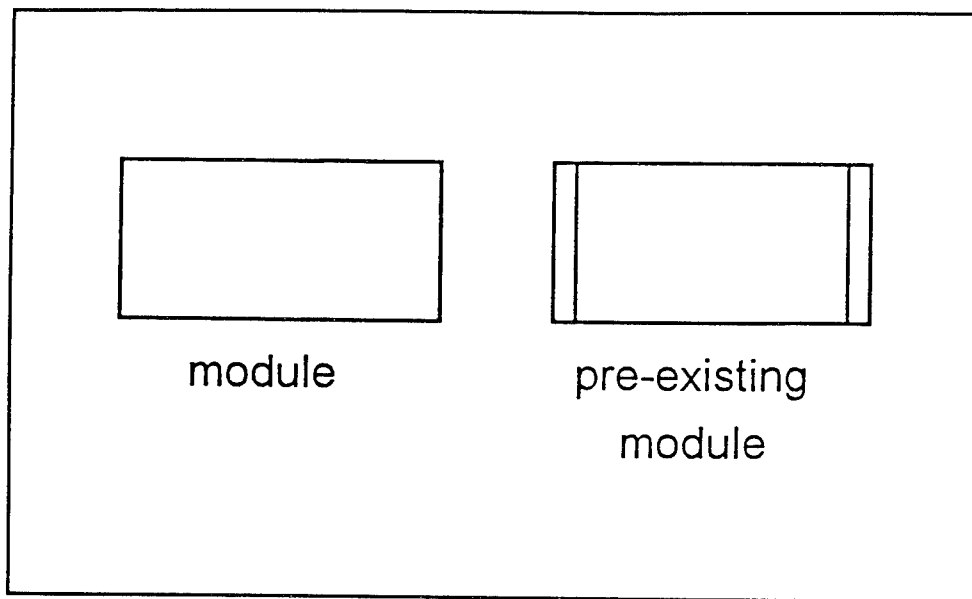
Figure 13. Module graphic representations
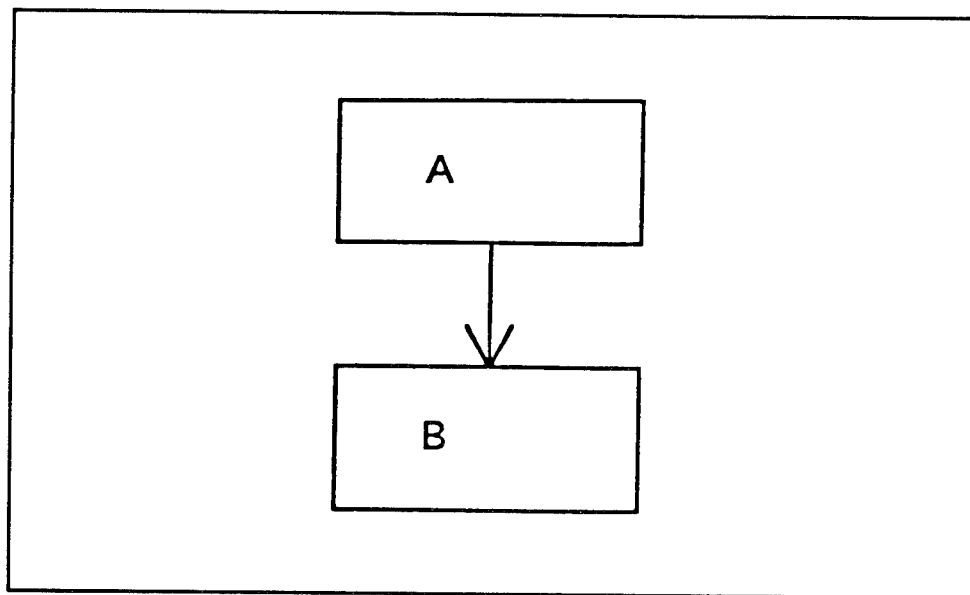
Figure 14. Connection between modules

Module Communications

The module communications occur through input or output data that flow between modules, and they are called input or output parameters. The arrow indicates the flow direction. The names or identification given for data flowing as parameters to and from subroutines are names as used in the calling module. Its usefulness distinguishes between parameters that are elements of

control as flags, errors, or end-of-data indicators. A small dot on the tail of any arrow indicates *control*, a small circle indicates *data*. Figure 15 shows the passage of both parameter types.
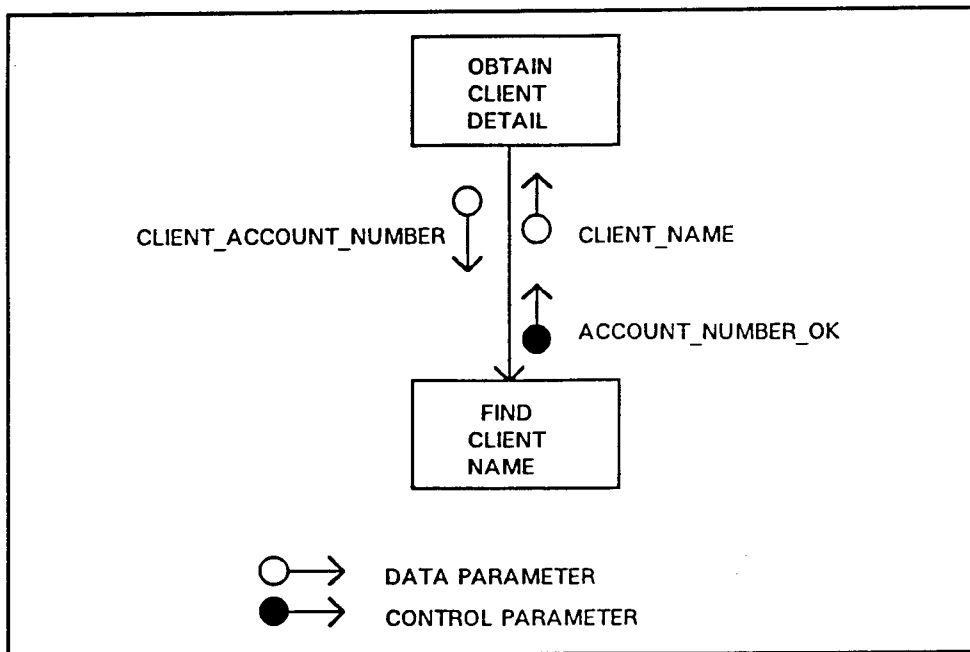
Figure 15. Module communications

Activation

Activation is a subroutine calling form and it can be:

● Unconditional.

A module calling is executed without a conditional test.

● Conditional.

A change in the logic flow execution occurs inside the module caused by a conditional test. Figure 16 shows a selection conditional test.

● Repeatedly.

A module calls one or more modules through an inner loop. Figure 17 shows the annotation used in this case, and it is important to observe that only the repeated modules may be encompassed.
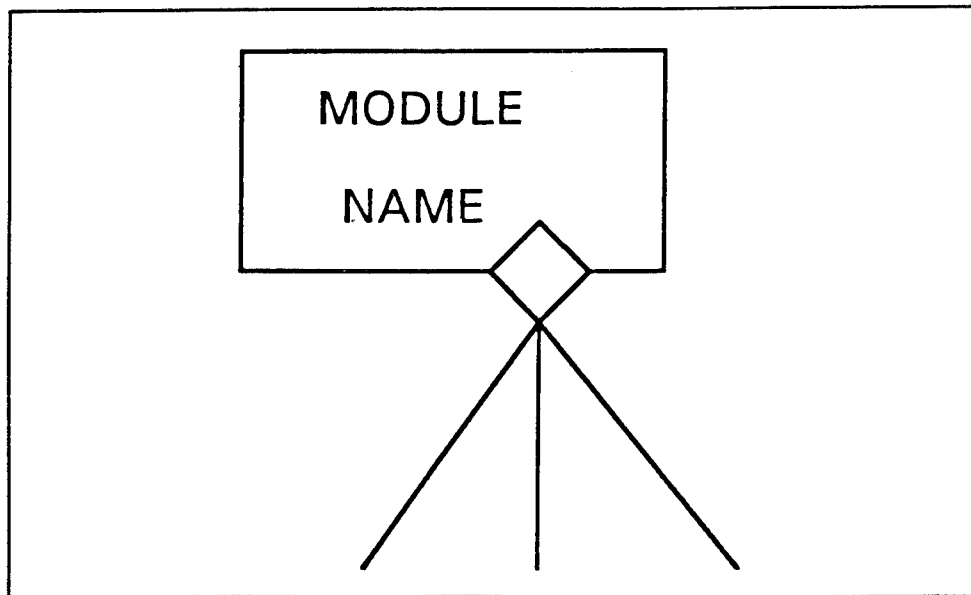
Figure 16. Conditional activation
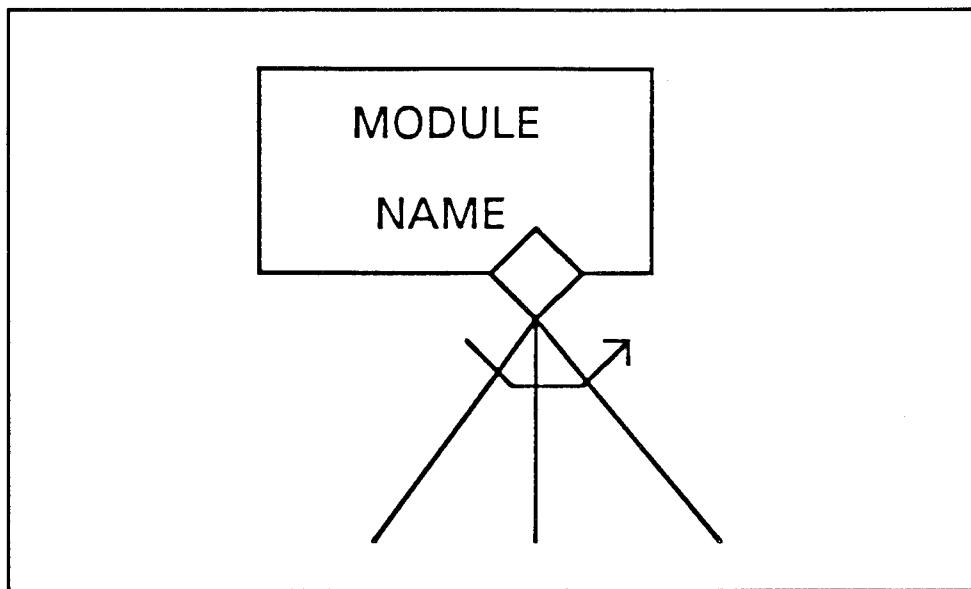
Figure 17. Repeated activation

Connector

Figure 18 shows the two connector types:

- Intern connection - repeated modules at same page.

- Extern connection - repeated modules between different pages.
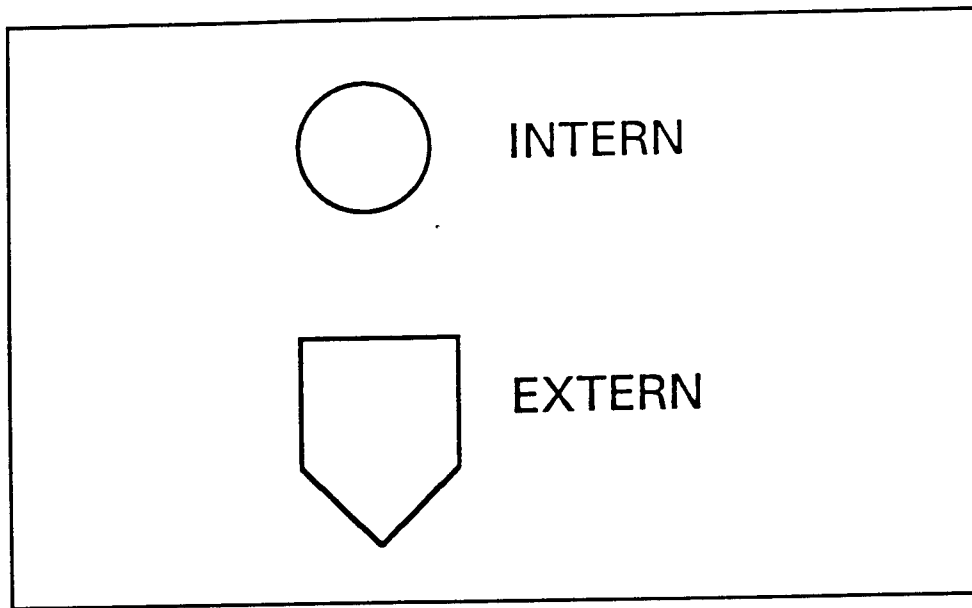
Figure 18. Connectors

5   Application

The Corps of Engineers National Dam Inventory Data Update Program was developed at the U.S. Army Engineer Waterways Experiment Station (USACEWES) with the objective to inventory all dams on Army bases that meet one of the following requirements:

- Greater than 6 ft high with a maximum water impounding capacity of at least 50 acre-ft.

- At least 25 ft with a maximum water impounding capacity in excess of 15 acre-ft.

In this chapter, a basic structured documentation for the dam inventory program will be developed to show that the E-R Diagram and Structure Chart are tools that are very easy to use and very powerful modeling tools.

E-R Diagram

As mentioned in Chapter 3, the E-R Diagram is a data modeling tool. Based on program files with the *dbf* extension, it was possible to construct the database E-R model that enabled an excellent data overview. Figure 19 shows the E-R Diagram modeling. The relationships aren't named because they are 1:N without their own attributes; therefore, they do not generate files and it is not obligatory to have identification.

The entities and their respectives keys and attributes are:

- Corps (#id, cwis, div, dist, owner, state, county, damname, lake, river, purpose, type, hydhgt, crest, maxcap, lat, long, seismic, lastinsp, nextinsp, yearcmpl, category, subcat1, subcat2, cat_comm, updtdate, startdate, schedcmpl, cngdst1, region, basin, nearcity, distance, population, strhgt, norcap, hazardcode, spilltype, spillwid, spilldis, volmat, numlocks, engr, const, sate2, lenlock, widlock, miscode, cost, eap, eap_i, eap_o, eap_n, eap_n1, eap_n2, eap_e, eap_e1, eap_e2, eap_comm, rel_well, rel_well1, rel_well2, drain, instr, instr_mon, water_qual, fed, fed_power, fed_prop, nfed,
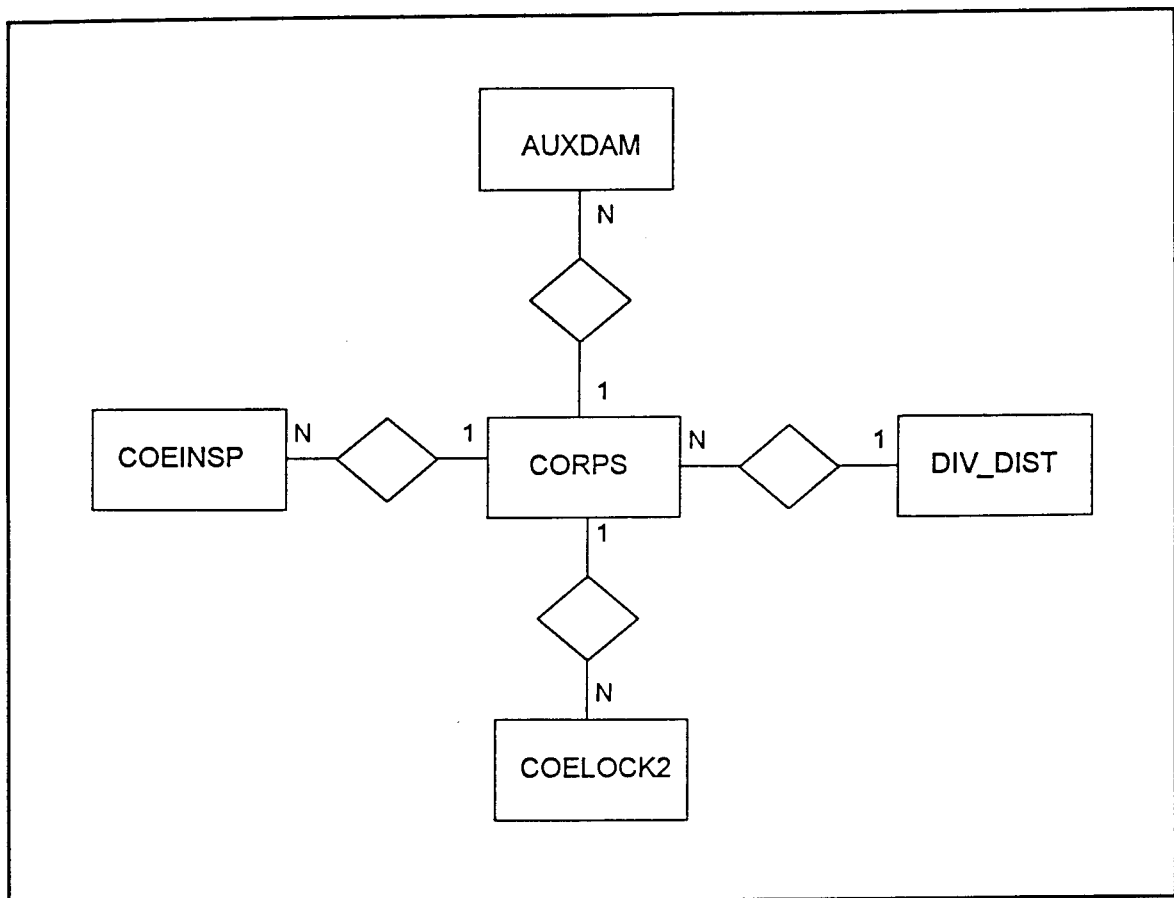
Figure 19. Application's E-R diagram

nfed_power, nfed_prop, fercno, licensee, total_insp, insp_freq, surf_area, drain_area, poc, phone).

- Auxdam (#id, auxname, auxtype, auxstrht, auxcrlen, auxvol).

- Coeinspd (#inspdate, #id).

- Div_dist (#div_code, #dist_code, division, district).

- Coelock2 (#id, loc_len2, loc_wid2).

The entities Auxdam and Div_dist have compound keys because more than one attribute is necessary to identify one element of the entities. It is not the objective of this report to explain the significance of all attributes.

Structure Chart

As mentioned in Chapter 4, the Structure Chart permits a creation of a hierarchy between the program modules.

Figure 20 shows the main structure of the Corps of Engineers National Dam Inventory Data Update Program. Figures 21 and 22 show details and how they are linked with the main structure through page connectors like C1, C2, CS, CI, and CM.

Table 1 describes all program functions and procedures with their purposes and parameters used.
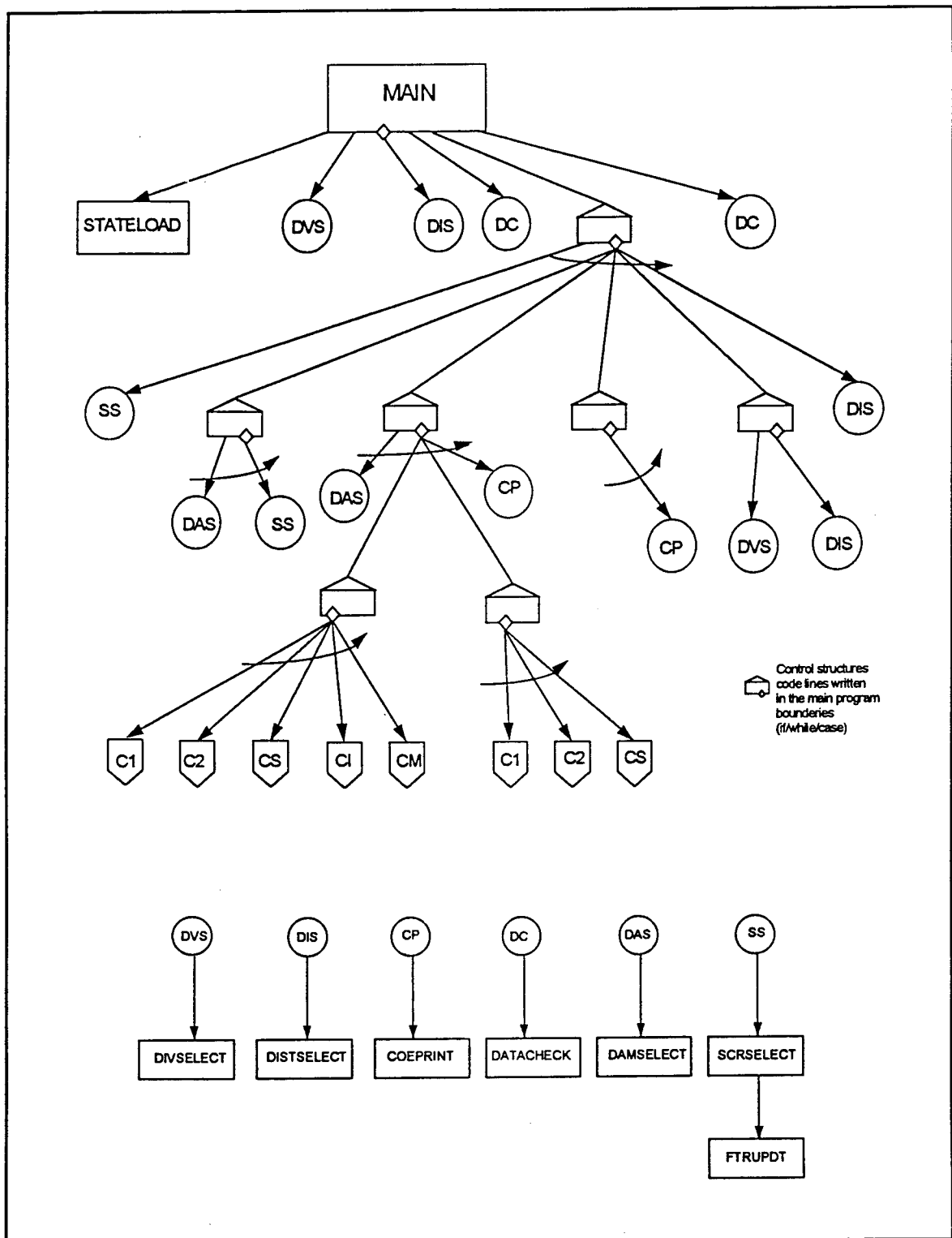
Figure 20. Main structure chart
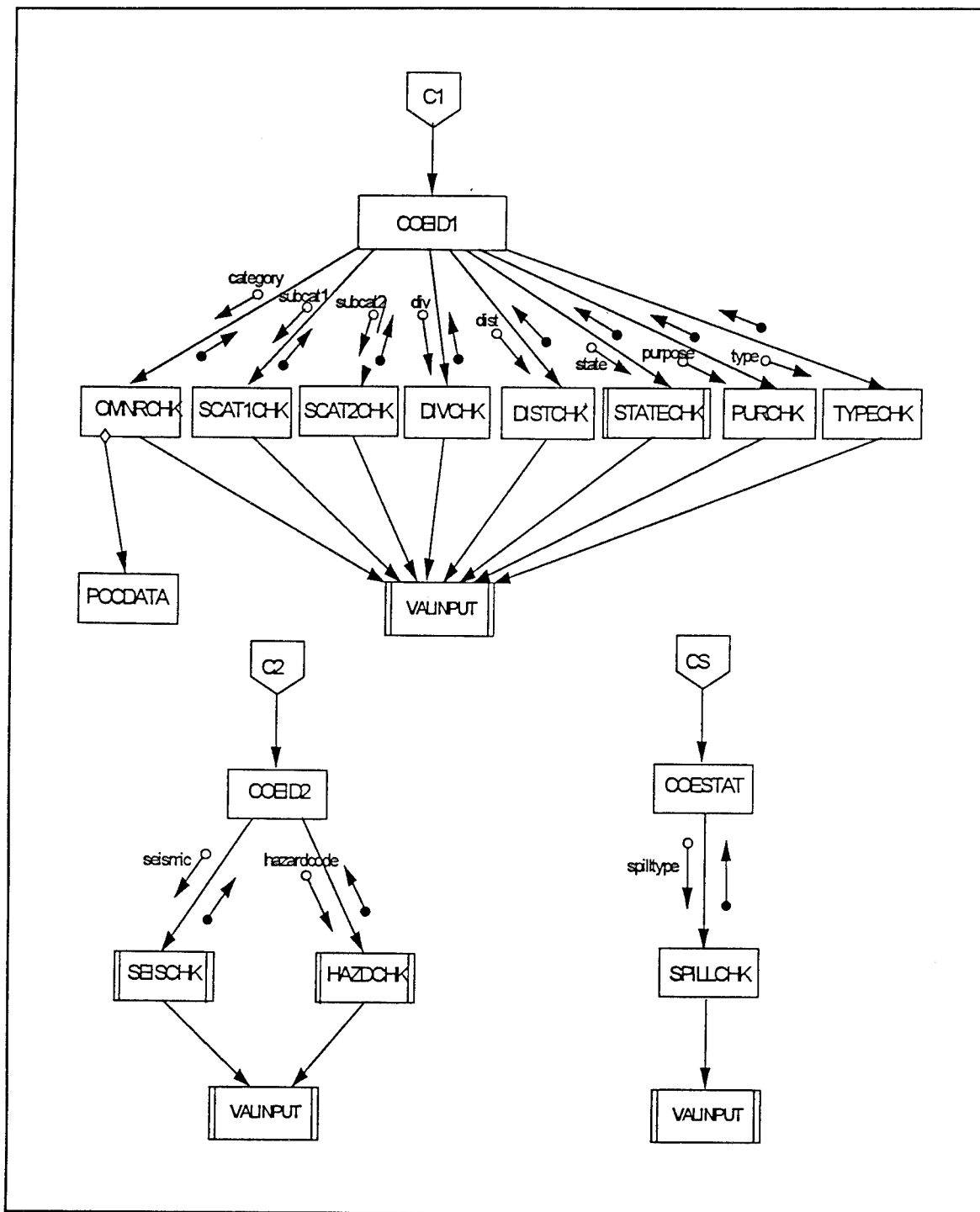
Figure 21. Structure chart with page connectors C1, C2, and CS
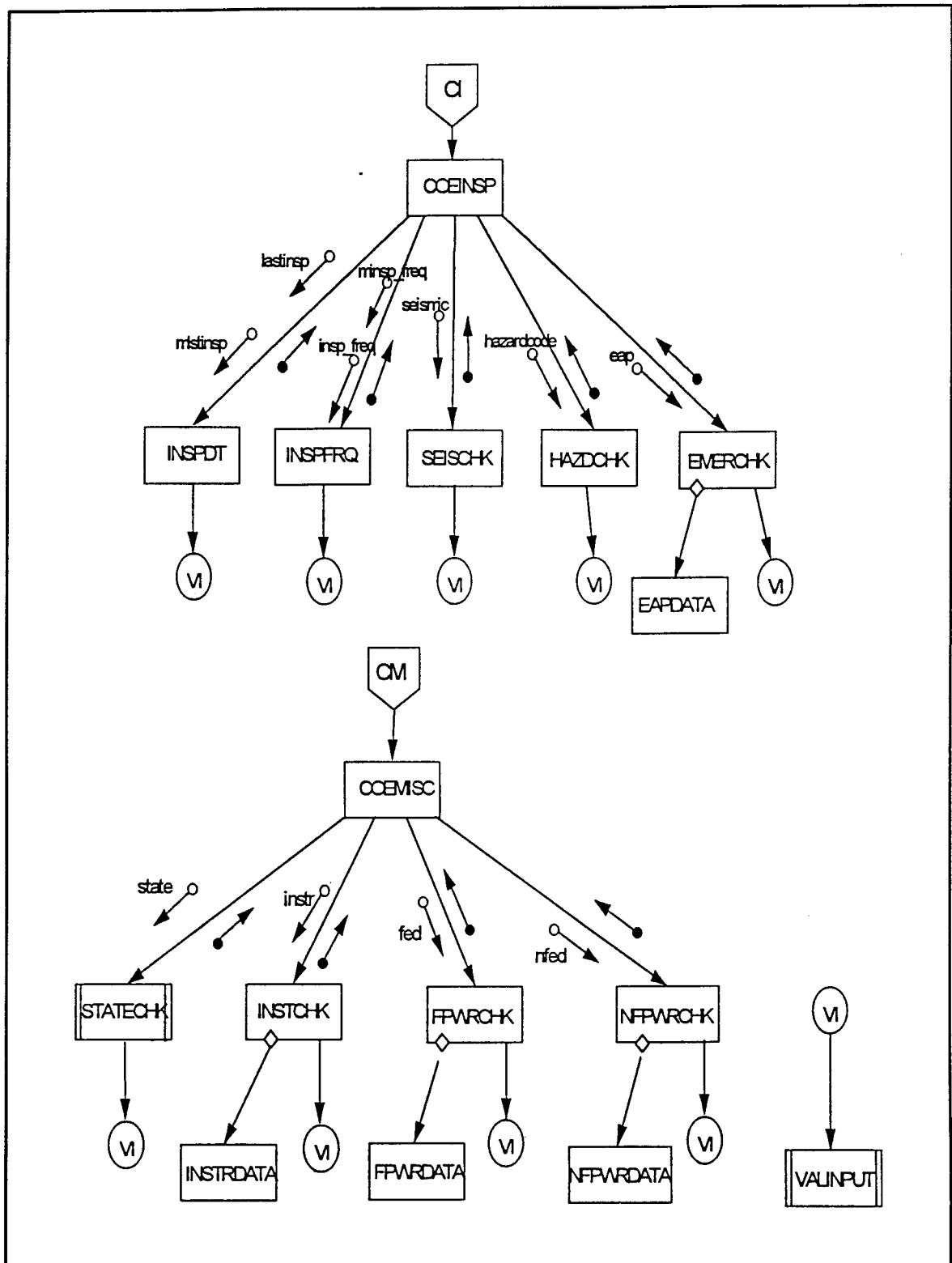
Figure 22. Structure chart with page connectors CI and CM

Table 1
Procedures and Functions of Application

Name	Purpose	Type	Parameters
COEID1	Screen 1 of COE dams data - id and location	Procedure	
COEID2	Screen 2 of COE dams data - id and location	Procedure	
COEINSP	Screen 4 of COE dams data - inspection and hazard	Procedure	
COEMISC	Screen 5 of COE dams data - miscellaneous	Procedure	
COESTAT	Screen 3 of COE dams data - statistics	Procedure	
DATACHEK	Check dam record	Procedure	
DAMSELEC	Select dam with cursor	Procedure	
DISTCHK	Validate usacedistrict	Logic function	Input_var
DISTSELECT	Select district with cursor	Procedure	
DIVCHK	Validate usace division	Logic function	Input_var
DIVSELECT	Select division with cursor	Procedure	
EAPDATA	Window routine for eap data	Procedure	
EMERCHK	Validate emergency action plan and prompt and display EAP data	Logic function	Input_var
FPWRCHK	Prompt and display data for Federal regulated power installation.	Logic function	Input_var
FPWRDATA	Window routine for Federal power data	Procedure	
FTRUPDT	Update auxiliary features data	Procedure	
HAZDCHK	Validate downstream hazard code	Logic function	Input_var
HELP	Provide aid and/or extra information to user and not represented in the structure chart	Procedure	
INSPDT	Replace nextinsp with new value	Logic function	New_var New_var2
INSPFRQ	Replace nextinsp with new value based on change in frequency	Logic function	New_var New_var2
INSTRCHK	Prompt and display instrumentation data	Logic function	Input_var
INSTRDATA	Window routine for instrumentation data	Procedure	
NFPWRCHK	Prompt and display data for non-Federal regulated power installation.	Logic function	Input_var
NFPWRDATA	Window routine for non-Federal power data	Procedure	
OWNRCHK	Validate type of owner	Logic function	Input_var
POCDATA	Window routine for poc data if category < > "Corps"	Procedure	
PURPCHK	Validate purpose of dam	Logic function	Input_var
SCAT1CHK	Validate subcategory No. 1	Logic function	Input_var
SCAT2CHK	Validate subcategory No. 2	Logic function	Input_var

(Continued)

Table 1 (Concluded)			
Name	**Purpose**	**Type**	**Parameters**
SCRSELECT	Determine the screen to display	Procedure	
SEISCHK	Validate seismic zone	Logic function	Input_var
SPILLCHK	Validate type of spillway	Logic function	Input_var
STATECHK	Validate state code	Logic function	Input_var
STATELOAD	Load state codes into an array (starr)	Procedure	
TYPECHK	Validate type of dam	Logic function	Input_var
VALINPUT	Validate data input	Procedure	

6   Improvements

This chapter presents a Corps of Engineers National Inventory Data Update Program improvement based on diagrams generated in Chapter 5. The first step of the improvement will be on the E-R Diagram, and the second step will be on the Structured Chart.

E-R Diagram

Comparing Figure 19 with Figure 23, it is possible to observe eight new entities introduced in the new E-R Diagram:

- EMPLAN is an emergency plan entity with two attributes as shown in the following tabulation. A file named EMPLAN.DBF was created with 2 characters as the length size of the CODEMPLAN field and 15 characters as the length size of the NAME field.

CODEMPLAN	NAME
Y	Yes
N	No
NR	Not Required
O	Other

- HAZARD is an entity with four attributes as shown in the following tabulation. A file named HAZARD.DBF was created with 1 character as the length size of the CATEGORY field, 12 characters as the length size of the NAME field, 15 characters for the LOSS_LIFE field, and 15 characters for the ECONO_LOSS field.

CATEGORY	NAME	LOSS_LIFE	ECONO_LOSS
L	Low	None Expected	Minimal
S	Significant	Few	Appreciable
H	High	More Than Few	Excessive
O	Other		
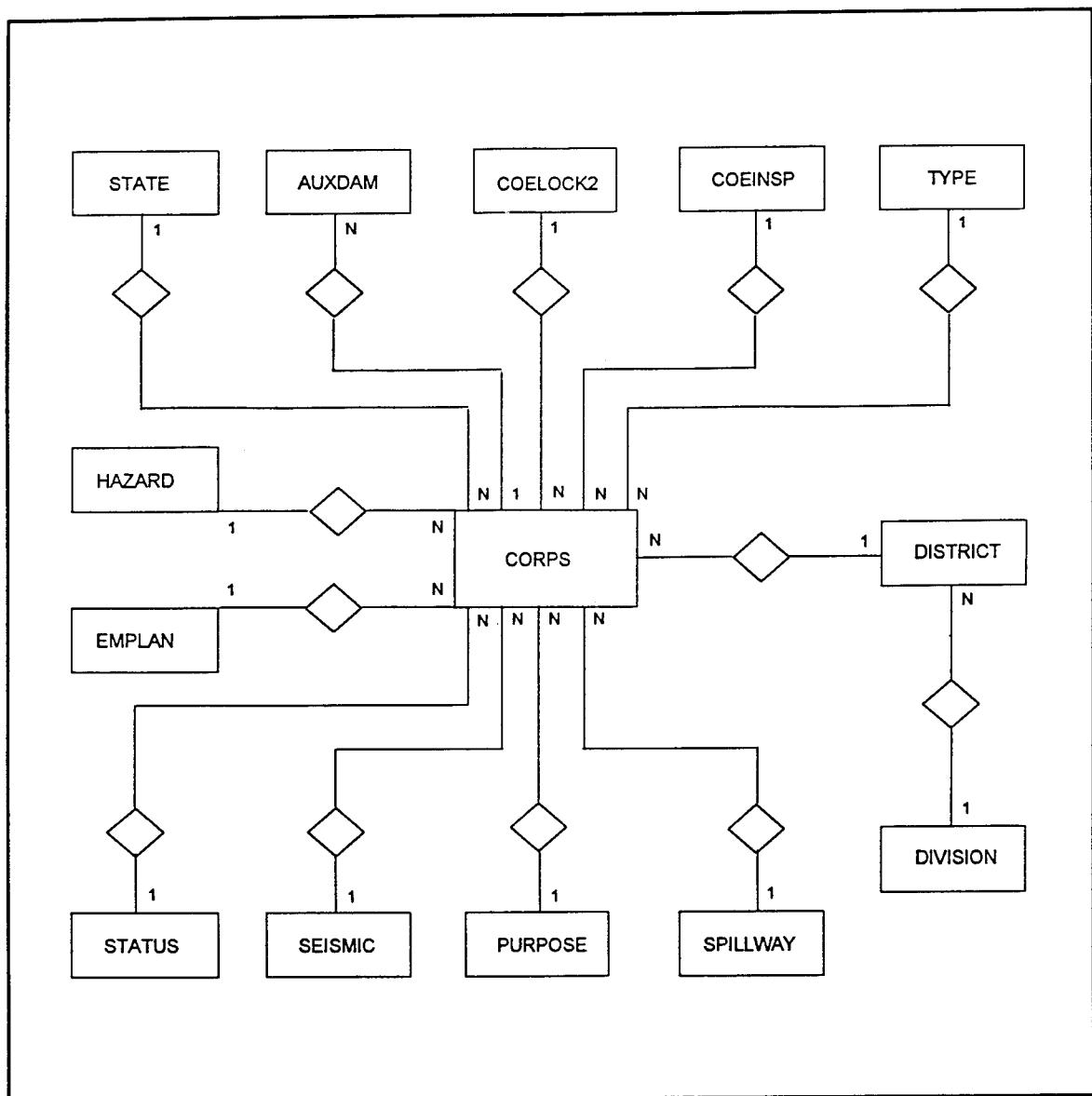
Figure 23. E-R diagram improved

- PURPOSE is an entity with two attributes as shown in the following tabulation.. A file named PURPOSE.DBF was created with 1 character as the length size of the CODPURPOSE field and 45 characters as the length size of the NAMEPURPOS field.

CODPURPOSE	NAMEPURPOS
I	Irrigation
H	Hydroelectric
C	Flood Control and Storm Water Management
N	Navigation
S	Water Supply
R	Recreation
P	Fire Protection, Stock, or Small Farm Pond
F	Fish and Wildlife Pond
D	Debris Control
T	Tailings
O	Other

● SEISMIC is an entity with three attributes as shown in the following tabulation. A file named SEISMIC.DBF was created with 1 character as the length size of the ZONE field, 10 characters as the length size of the DAMAGE field, and 5 characters for the COEFFICIENT field.

ZONE	DAMAGE	COEFFICIENT
0	None	0.0000
1	Minor	0.025
2	Moderate	0.050
3	Major	0.100
4	Great	0.150

● SPILLWAY is an entity with two attributes as shown in the following tabulation. A file named SPILLWAY.DBF was created with 1 character as the length size of the TYPE field and 15 characters as the length size of the NAME field.

TYPE	NAME
C	Controlled
U	Uncontrolled
N	None
O	Other

● STATUS is an entity with two attributes as shown in the following tabulation. A file named STATUS.DBF was created with 1 character as the length size of the CODSTATUS field and 30 characters as the length size of the NAMESTATUS field.

CODSTATUS	NAMESTATUS
D	Corps Designed
C	Corps Constructed
O	Operational
F	Regulated for Flood Control
N	Regulated for Navigation

- TYPE is an entity with two attributes as shown in the following tabulation. A file named TYPE.DBF was created with 2 characters as the length size of the CODTYPE field and 15 characters as the length size of the NAMETYPE field.

CODTYPE	NAMETYPE
RE	Earthfill
ER	Rockfill
PG	Gravity
CB	Buttress
VA	Arch
MV	Multi-arch
CN	Concrete
MS	Masonry
ST	Stone
TC	Timber Crib
OT	Other

- STATE is an entity with two attributes as shown in the following tabulation. A file named STATE.DBF was created with 2 characters as the length size of the CODSTATE field and 15 characters as the length size of the NAME field.

The entities described were created because they have their own attributes. One relationship with CORPS can be established, and a maintenance file can be realized through DataBaseUtility(DBU) that enables insert, edit, delete, or filter data. As an example, the original program presents an array with three disadvantages of this solution described as:

- RAM memory allocation because the array remains in the main memory during run time.

- No simple reutilization array because with the file STATE, it can be easily reutilized by any other program.

- Difficult maintenance because if a new state is created or if it is necessary to restrict a state, the insert or the filter can be easily and very quickly realized through the DBU without change code.

A second example is related with the creation of the SEISMIC file. This entity has three attributes: zone, damage, and coefficient, and one file can be created. A new zone creation or a coefficient change in the file through DBU is easier than changing program code lines, and the file is available to a future reutilization in another program. All the other files created have the same justification.

Figure 19 shows two distinct entities, DIVISION and DISTRICT. A relationship between the entities was established through a foreign key so that the stored data quantity decreased because the division name and code were repeated for all districts subordinated. It is important to emphasize that

Division and District are different entities with owner attributes. The correspondent files are DIVISION.DBF(#CODDIV, NAMEDIV) and DISTRICT.DBF (#CODDIST, NAMEDIT,CODDIV).

This proposal splits the file CORPS.DBF at 11 files so that each file corresponds to one Division. The CORPS.DBF has approximately 400 Kb and, with the split process, the larger file occupies approximately 120 Kb of secondary memory. This enables a faster database process. New files are listed:

LMV.DBF	Lower Mississippi Valley Division
MRD.DBF	Missouri River Division
NAD.DBF	North Atlantic Division
NCD.DBF	North Central Division
NPD.DBF	North Pacific Division
ORD.DBF	Ohio River Division
SAD.DBF	South Atlantic Division
SPD.DBF	South Pacific Division
NED.DBF	New England Division
POD.DBF	Pacific Ocean Division
SWD.DBF	Southwestern Division

Structure Chart

Two improvements are suggested to the structure charts shown in Figures 20, 21, and 22.

- Decrease the control structure (if/while/case) levels as shown in Figure 20 through new modules called by the main module so that a better modular program structure may be developed.

- Packing the program at more than one source font for easier maintenance.

Control Structure Levels Decreasing

The level number as shown in Figure 20 was decreased, and a better modular structure is proposed. Figure 24 shows the new structured charts for the Corps of Engineers National Dam Inventory Data Update Program. Table 2 shows comparisons between the old structure and the new one, with advantages of the main new structure.

It is possible to list any new modular structure consequence:

v

- The public variables minor number permits using a less RAM, and local variables were used because they only are allocated in RAM when the procedure where they are defined is allocated.

- No use of unconditional deviation command, like the GOTO command, because it isn't a well-structured programming technique.

- A small main program enables initialization procedures and main select menu.

- Approximately 30-percent line code reduction.

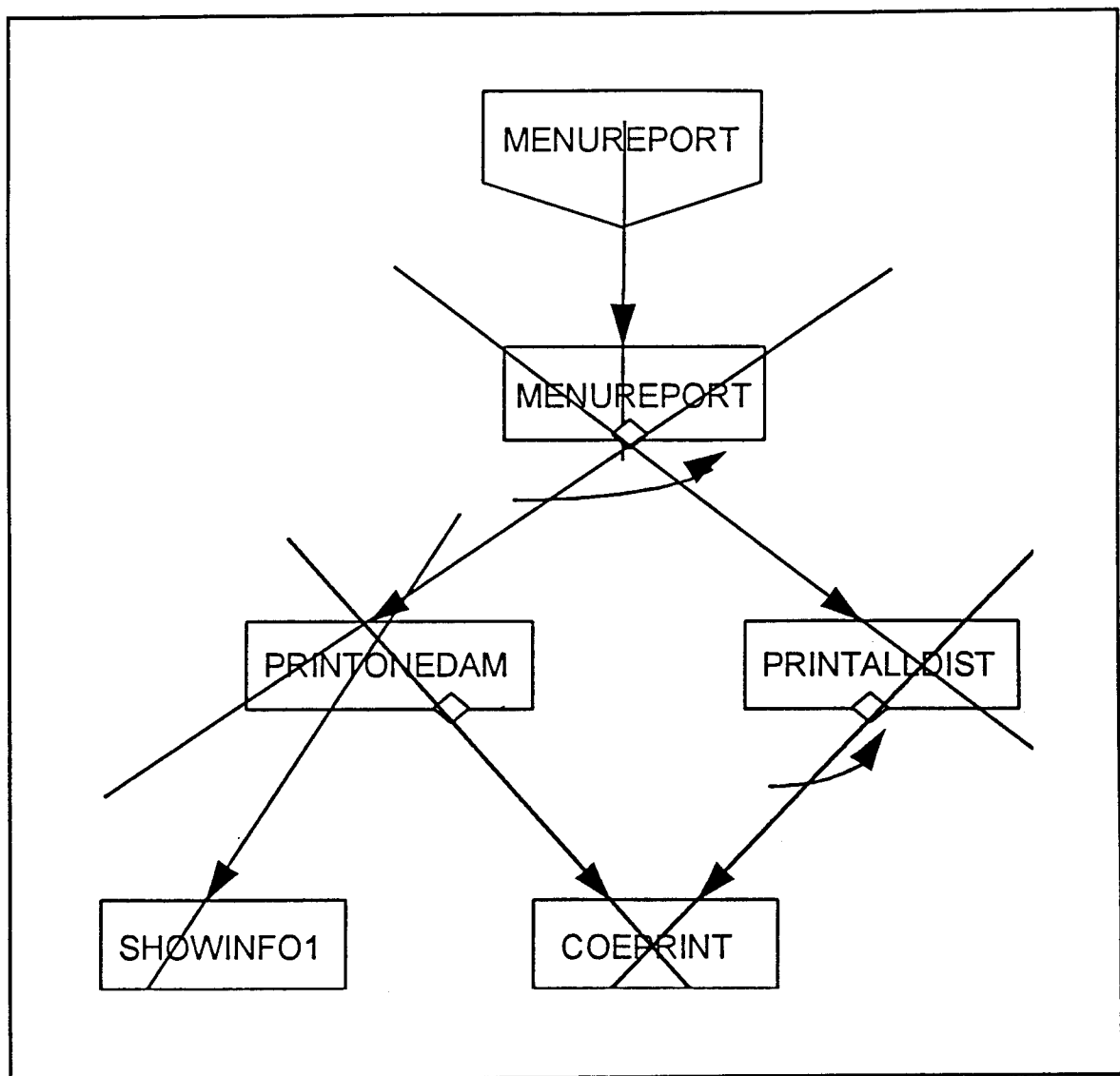- Reduction from 8 to 1 control structure in the main program allows a better maintenance.

Figure 24. New structure charts for National Dam Inventory Update Program

Table 2			
Old and New Structure Program Comparisons			
Item	Old Structure	New Structure	Advantages
Main program size	2,550 lines	42 lines	- Easy maintenance
Main program control structures (if/while/case/for)	8	1	- Easy maintenance - Better modular structure
Programs (PRG extension)	CDAMS (2,550 lines) COEPRINT (916 lines) COEINDEX (42 lines)	DAMCE (42 lines) DAM (639 lines) DAMBEG (70 lines) DAMDIV (69 lines) DAMDBSE (114 lines) DAMPRINT (1,007 lines) DAMLIB (370 lines) DAMMNT(50 lines)	- Easier development - Easier maintenance - Code reutilization
Total Program Lines	3,508	2,361	- Less secondary memory
Main file index time[1]	10 s	3 s	- Faster
GOTO Commands	6	0	- Unconditional deviation is not a good structure command
Global variables	10	2	- Using less RAM
Arrays	2	0	- Using less RAM
Functions	Add Edit Report	Add Edit Report Database Consult Delete Index maintenance	- More functionality

[1] 486 SX/IBM Processor

Program Packing

The Corps of Engineers National Dam Inventory Data Update main file has 2,550 code lines that require a hard program maintenance when an alteration is necessary. This difficulty is because the first task will be to locate the function needing alterations, and the second one will be to recompile 2,550 code lines. Packing is a technique that enables the structured chart to be divided so that the modules, procedures, or functions, can be arranged as needed at source files. This process enables source files to be generated with specific objectives. For example, a source file called DAMPRINT.PRG combines the procedures that generate reports. It is possible to run a disconnected file test. After a successful run, the source files can be combined into one executable program.

Figure 25 shows the source files to the proposed packing that are:
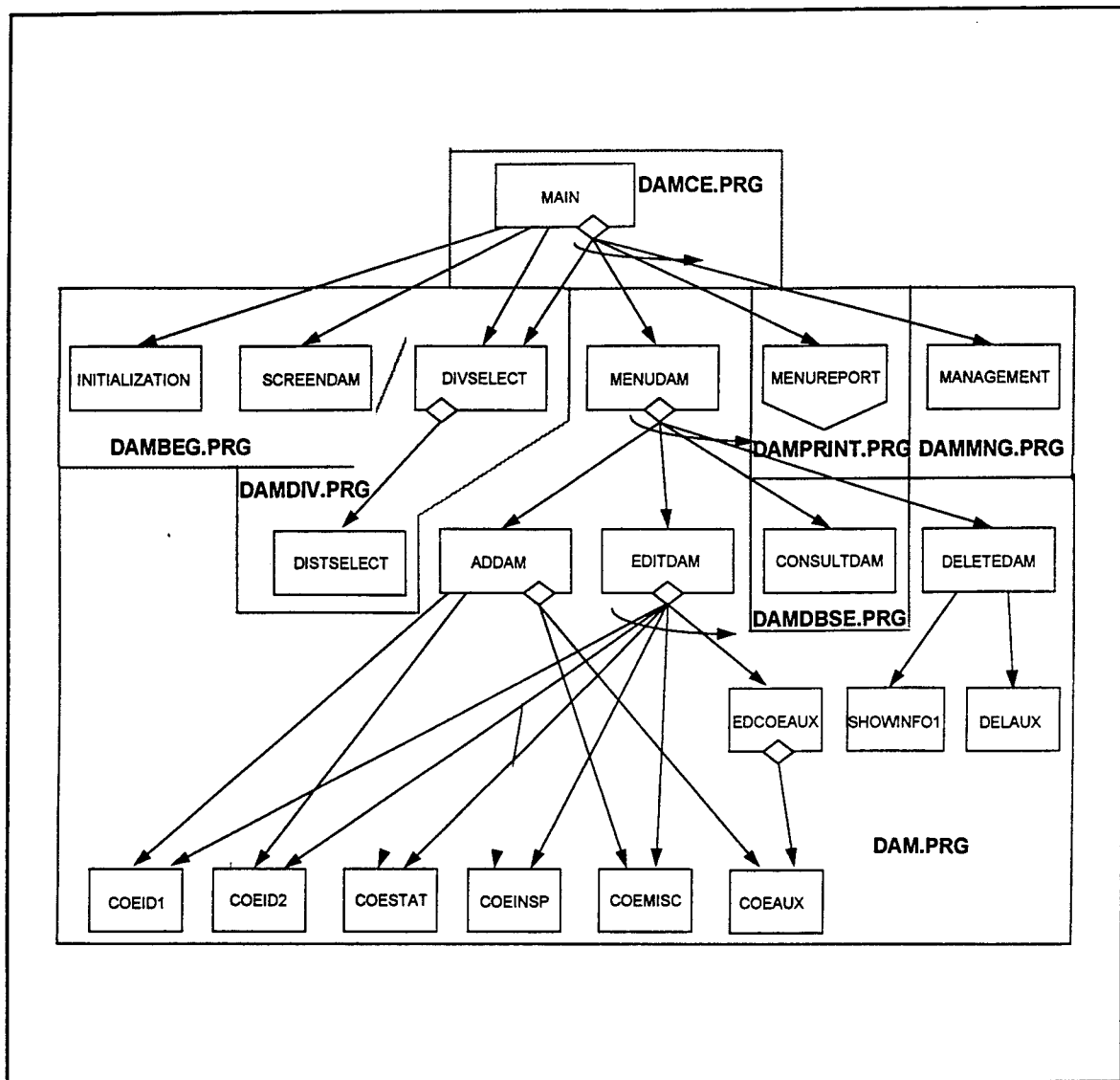
Figure 25. Proposed program packing source files

- DAMCE.PRG, main source file.

- DAMBEG.PRG, with initialization procedures.

- DAMDIV.PRG, with Division and District select procedures.

- DAM.PRG, with add, edit, and delete procedures.

- DAMDBSE.PRG, with TBrowse consult procedure.

- DAMPRINT.PRG, with report procedures.

- DAMMNG.PRG, with management procedures like index file process.

- DAMLIB.PRG, with support functions like a message sending.

The file number might indicate a maintenance problem, although it is not. If a programmer has a report problem, he selects the DAMPRINT.PRG file, makes the changes, compiles it separately, and later links it with the others files. The modular structure enables a better code reutilization as well. The file DAMLIB.PRG is an example that contains all generic use procedures and functions that can be reutilized in other programs. Another example is the MENUPRINT procedure located in the DAM.PRG file. A report menu can be generated, and with few adaptations, it is possible to generate another specific menu.

7   Conclusion

The E-R Diagram and Structured Chart are easy-to-use tools, and they permit an excellent overview of the data and functions program. Their use enables us to create a minimum documentation with the following advantages:

- A constant program structure overview.

- A program development with suggestions for a better structure.

- An excellent database and function graphical interface between user and programmer that permits participation in the program development.

- A simple maintenance for program life by all users.

- A database and/or functions reutilization in others programs.

A suggested life cycle to develop structured programs follows:

Analysis Phase Activities:

- List objectives of the system.

- Study problem defined in the objectives.

- Define DATABASE using the E-R Diagram (if necessary).

Project Phase Activities:

- Program structure charts design.

- Program specifications in the language to be used in the system development.

- Screen and report designs.

Implementation

- Program codification in a computer.

Operation

• System installation and maintenance.

This work developed a comparison between the original Corps of Engineers National Dam Inventory Data Update Program and a new version. The two programs work well and the comparative time data presented in Table 2 are of little consequence, because the difference is small and can be decreased by a more rapid processor. The source program sizes are not important because only a few bytes are economized. The most important difference in this work is the program documentation proposed through the E-R Diagram and Structured Chart. The E-R Diagram is not always necessary because some scientific programs do not require database files, but the Structured Chart is applicable for all programs.

Resuming all this work, the E-R Diagram and Structured Chart enable a better database and program MAINTENANCE, REUTILIZATION, and DOCUMENTATION.

References

Chen, Peter. (1990). *Data Base Management*, McGraw-Hill, New York.

Date, C. J. (1991). *Introdução a Sistemas de Bancos de Dados*. Editora Campus.

Edris, Earl V., Vanadit-Ellis, Wipawi, and Woo, Kay. (1991). *Department of the Army Inventory of Dams: Data Base Summary and User's Guide*, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.

Korth, Henry F., and Silberschatz, Abraham. (1991). *Sistemas de Bancos de Dados*. Editora Campus.

McMenamim, Stephen M., and Palmer, John F. (1984). *Essential Systems Analysis*. Prentice-Hall, Englewood Cliffs, NJ.

Page-Jones, Meilir. (1988). Projeto Estruturado de Sistemas. McGraw-Hill, New York.

Setzer, Valdemar W. (1986). *Banco de Dados*. Edgard Blücher.

Van Griethuysen, J. J. (1982). ISO TC97/SC5/WG3 CONCEPTUAL SCHEMA.

Yourdon, Edward. (1988). *Modern Structured Analysis*. Prentice-Hall, Englewood Cliffs, NJ.

Yourdon, Edward. (1989). *Structured Walk throughs*. Prentice-Hall, Englewood Cliffs, NJ.

Yourdon, Edward, and Constantine, Larry L. *Structured Design*. Prentice-Hall, Englewood Cliffs, NJ.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 1995	3. REPORT TYPE AND DATES COVERED Final report

4. TITLE AND SUBTITLE
Basic Structured Documentation to the Corps of Engineers National Dam Inventory Data Update Program Based on the E-R Diagram and Structure Chart

5. FUNDING NUMBERS

6. AUTHOR(S)
Alberto Tavares da Silva

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Engineer Waterways Experiment Station
3909 Halls Ferry Road, Vicksburg, MS   39180-6199

8. PERFORMING ORGANIZATION REPORT NUMBER
Miscellaneous Paper GL-95-3

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S Army Corps of Engineers
Washington, DC   20314-1000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The Corps of Engineers National Dam Inventory Data Update Program (NDI) is a database program developed through the Computer Applications in Geotechnical Engineering Program. The NDI is used to store and retrieve a variety of information pertaining to dams in which the Corps of Engineers has an interest. This report describes the purpose of the NDI and proposes a restructuring for the NDI to provide greater efficiency and ease of maintenance. Two basic tools for program development are described: the Entity-relationship (E-R) Diagram and the Structure Chart. Examples of the use of the E-R Diagram and the Structure Chart are presented in the report and examples using modules from the NDI illustrate their use in program development. The report provides a structured documentation and describes some improvements to the NDI based on the use of these tools. A life cycle for structured program development, including maintenance, is suggested and comparisons between the original and restructured versions of the NDI are presented. The proposed modifications allow programmers to quickly grasp the structure of the NDI and initiate improvements or modifications as necessary. Other advantages of the proposed modifications include faster operation and easier maintenance.

14. SUBJECT TERMS
Dam inventory
Dam safety
Databases
Entity-relationship diagram
Structure chart

15. NUMBER OF PAGES
46

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED		